Федеральное государственное автономное образовательное учреждение
высшего образования
Национальный исследовательский университет "Высшая школа экономики"

Дисциплина «Основные методы анализа данных»

Отчет о проделанной работе на тему “Анализ данных недвижимости”

Выполнил: студент группы БПМИ229
Доан Тай Ле Минь
Преподаватель: Миркин Борис Григорьевич

Москва, 2024

Оглавление

Введение

Наш выбранный датасет называется “Sleep and Health Metrics” . Этот набор данных показался оптимальным для анализа данных, так как он содержит достаточное количество признаков - 9, что не слишком много и не слишком мало.

Далее приведена общая структура данных.
Количество анализируемых объектов: 1000
Количество признаков: 9
Список признаков:

- **“Heart Rate Variability”**: Имитируемая изменчивость временных интервалов между ударами сердца
- **“Body Temperature”**: Искусственно созданная температура тела в градусах Цельсия
- **“Movement During Sleep”**: Синтетические данные о количестве движений во время сна
- **“Sleep Duration Hours”**: Общее количество часов сна, полученных в результате моделирования.
- **“Sleep Quality Score”**: Синтетический показатель, отражающий качество сна.
- **“Caffeine Intake (mg)”**: Количество имитируемого потребления кофеина в миллиграммах
- **“Stress Level”**: Индекс смоделированных уровней стресса.
- **“Bedtime Consistency”**: Имитация последовательности выполнения распорядка дня перед сном. Шкала от 0 до 1, где более низкие значения указывают на большую непоследовательность
- **“Light Exposure Hours”**: Искусственное время пребывания на свету в течение дня. Отражает типичное время пребывания на свету в дневное время.

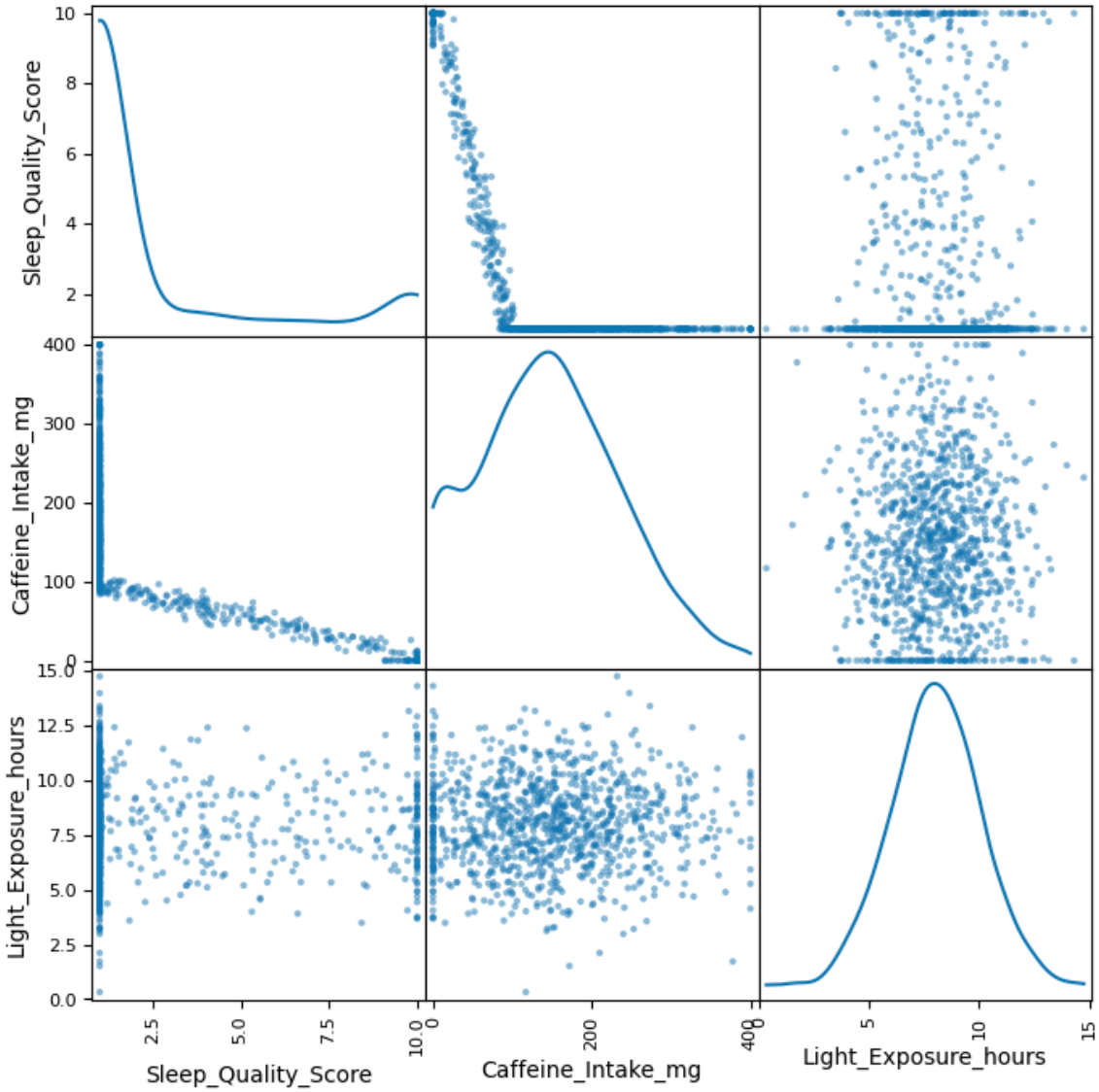
Ссылка на датасет:
https://www.kaggle.com/datasets/uom190346a/sleep-and-health-metrics/data

Фрагмент таблицы данных:

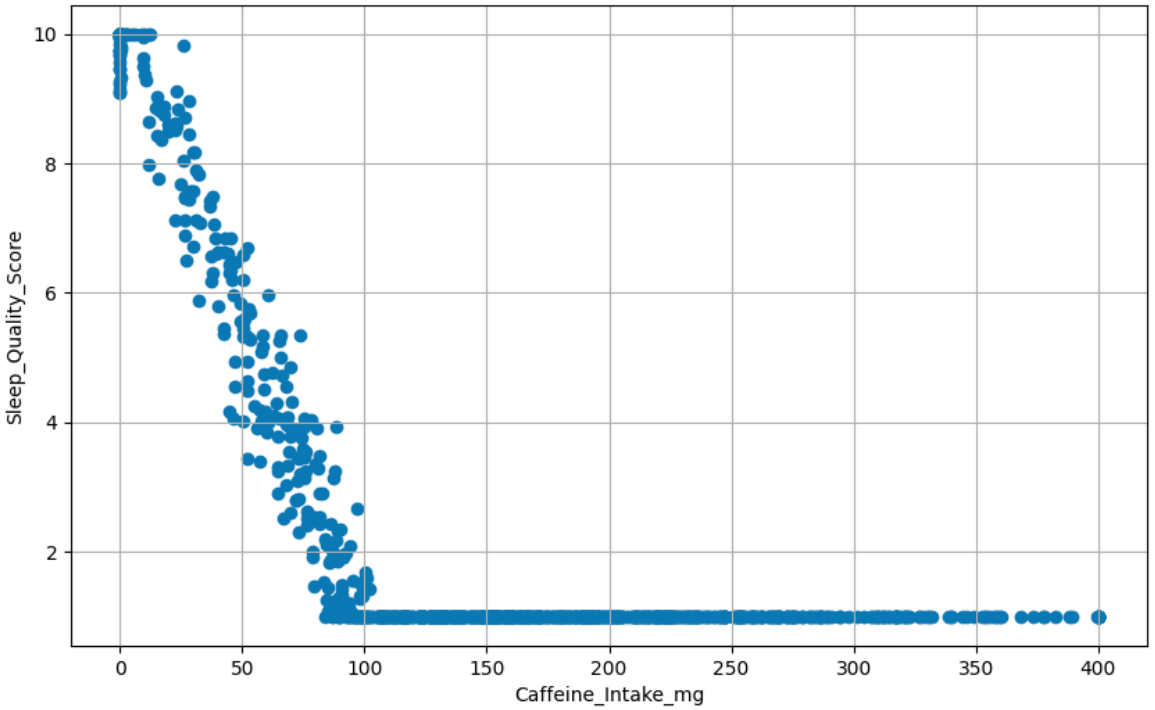
	Heart_Rate_Variability	Body_Temperature	Movement_During_Sleep	Sleep_Duration_Hours	Sleep_Quality_Score	Caffeine_Intake_mg	Stress_Level	Bedtime_Consistency	Light_Exposure_hours
0	79.934283	37.199678	1.324822	4.638289	1.000000	107.624032	2.771837	0.657037	7.933949
1	67.234714	36.962317	1.855481	6.209422	1.000000	104.658589	3.738138	0.144464	6.992699
2	82.953771	36.529815	1.207580	6.879592	10.000000	0.000000	3.115880	0.642949	7.655250
3	100.460597	36.176532	1.692038	10.331531	1.000000	116.990981	3.904008	0.453255	9.429463
4	65.316933	36.849112	0.106385	8.334830	1.000000	223.282908	4.571699	0.641492	10.555713
5	65.317261	36.696743	2.213294	5.496778	7.122153	22.576788	6.674309	0.655386	9.140975
6	101.584256	36.947597	2.001205	8.229054	1.000000	254.848265	4.357682	0.821583	8.203446
7	85.348695	36.817586	1.182911	5.179044	1.000000	198.777482	1.828749	0.770691	10.996024
8	60.610512	37.024776	2.659246	9.124037	2.409237	76.576668	7.280136	0.769314	7.374327
9	80.851201	36.232382	2.937570	6.793313	1.000000	135.847036	3.325820	0.788734	10.061839

Линейная регрессия и коэффициент корреляции

Для начала нужно найти два признака в данных с более или менее «линейным» полем рассеяния. Для этого были построены scatter графики для каждой пары признаков среди Sleep_Quality_Score , Caffeine_Intake_mg , Light_Exposure_hours .



Можно заметить, что признаки Caffeine_Intake_mg и Sleep_Quality_Score имеют более-менее линейную зависимость. Выберем в качестве целевой переменной Sleep_Quality_Score , а в качестве признака Caffeine_Intake_mg. Рассмотрим график для этих двух признаков поподробнее:



Было построено уравнение линейной регрессии

$$y = a * x + b$$

и получены следующие коэффициенты:

Regression coefficient: -0.02288

Intercept: 5.98460

Заметим, что коэффициент регрессии получился отрицательный, это означает что между признаками обратная зависимость. Более того, заметим, что Caffeine_Intake_mg принимает значение от 0 до 400, а Sleep_Quality_Score принимает от 0 до 10, что объясняет почему коэффициент регрессии так мал по модулю(на очень большое изменение признака Lot Caffeine_Intake_mg приходится малое изменение Sleep_Quality_Score).

Далее у нас значения коэффициентов корреляции и детерминации:

Correlation coefficient: -0.722

Determination coefficient: 0.521

Проинтерпретируем величину коэффициента детерминации. Она больше нуля, это значит, что наше регрессионное уравнение предсказывает лучше константного предсказания. А еще она немного больше 0.5, это значит, что наше регрессионное

уравнение не идеально, но приемлемо. Через scatterplot выше видно, что после того, как Caffeine_Intake_mg принимает значение больше 100, у Sleep_Quality_Score только значение 1. Это объясняет почему коэффициент детерминации только чуть выше 0.5. Но если мы срежем, чтобы у Caffeine_Intake_mg только значение до 100, то коэффициент детерминации будет **0.964**, что очень хорошо, модель выдает почти идеальные ответы.

Посмотрим на предсказания для 4-х случайно выбранных объектов:

	x_ix	y_true_ix	y_pred_ix
0	0.000000	10.0	5.984595
1	116.990981	1.0	3.308270
2	104.658589	1.0	3.590390
3	107.624032	1.0	3.522551

Как видно, все 3 объекта плохо предсказаны, что и следовало из коэффициента детерминации.

И наконец, анализируем среднюю относительную ошибку регрессионного уравнения на всех объектах таблицы данных как в парадигме анализа данных, так и парадигме машинного обучения:

Error_DA = 1.16(116%) **Error_ML** = 1.55(155%)

Понятно, что модель плохо предсказывает, и это ожидаемо по величине коэффициента детерминации и согласуется с ней. Как выше объясняли, количество объектов, у которого значения Caffeine_Intake_mg больше 100(~ 700 из 1000) значительное. Поэтому, если у нас модель выберем Caffeine_Intake_mg только значение до 100, то получим:

Error_DA = 0.15(15%) **Error_ML** = 0.14(14%)

Это согласуется с коэффициентом детерминации 0.964 и можно тогда убедиться, что модель хорошо работает, ошибка получается совсем небольшая.

Метод главных компонент

Нами было выбрано три следующих признака:

Heart_Rate_Variability: Изменчивость временных интервалов между ударами сердца

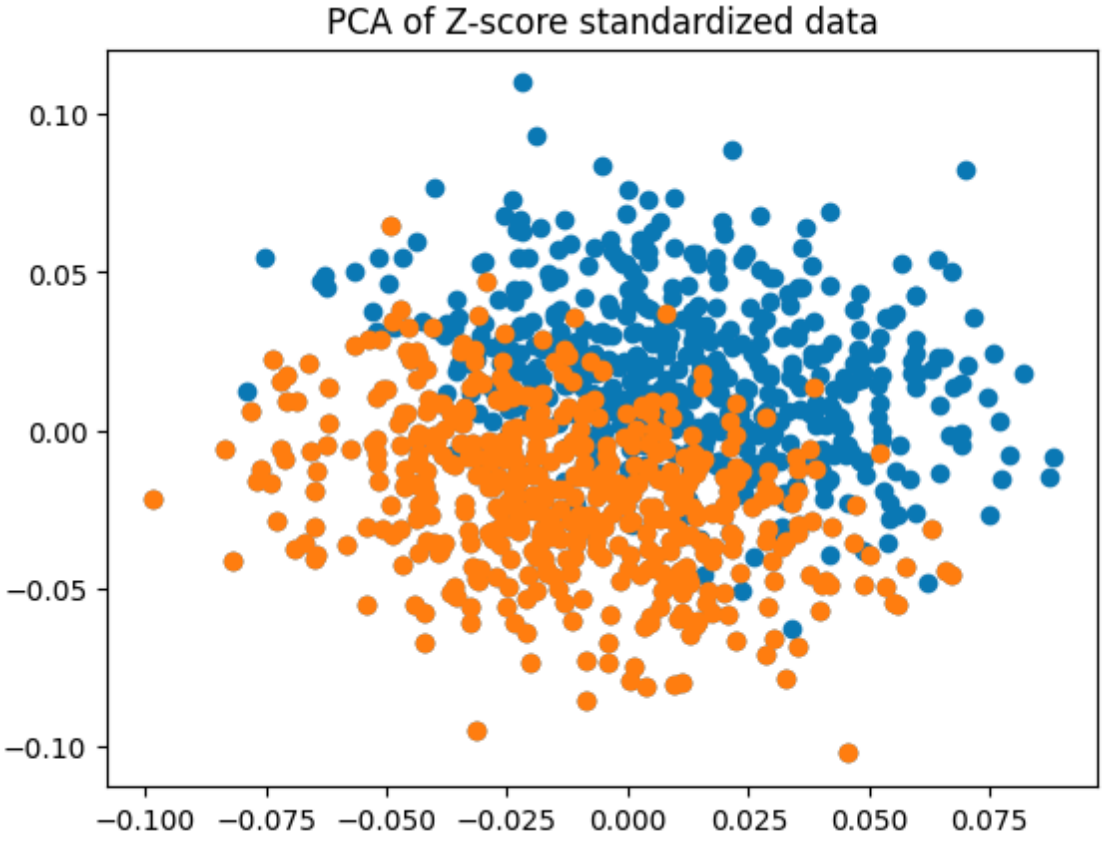
Body_Temperature: Температура тела в градусах Цельсия

Stress_Level: Индекс уровней стресса

Все эти признаки являются показателями здоровье человека.

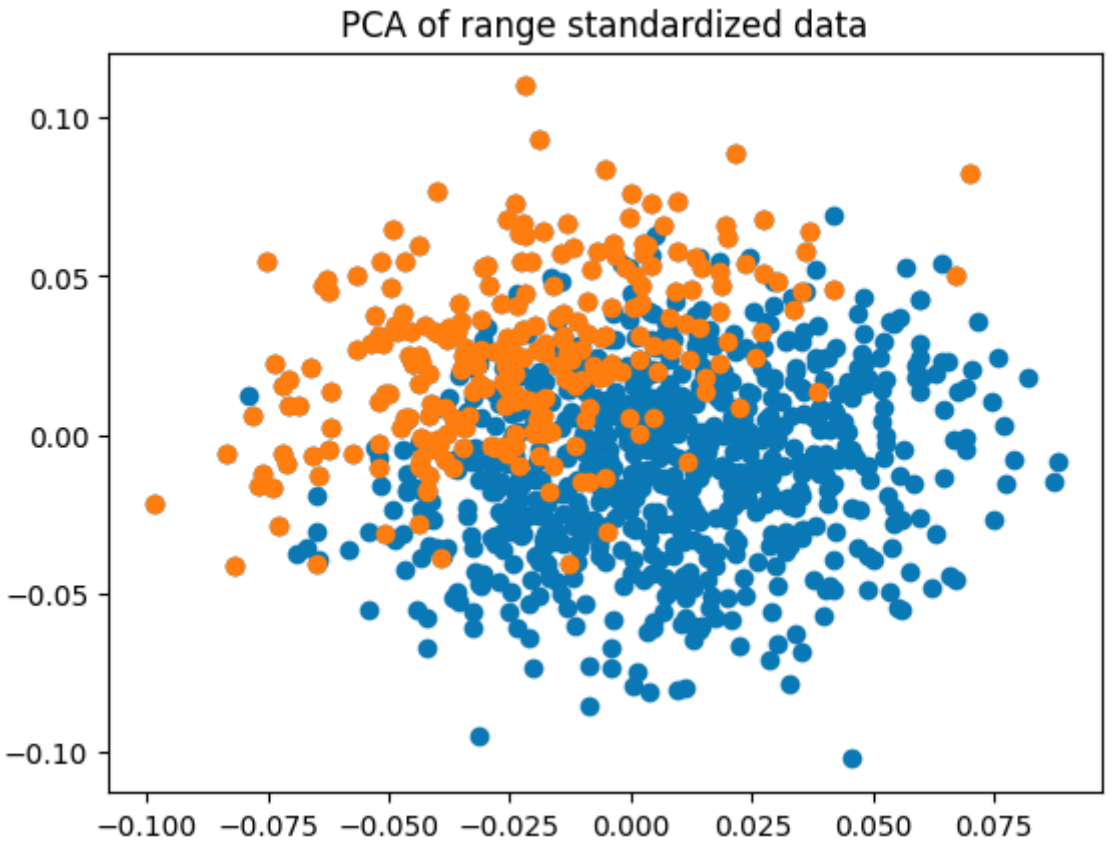
Сначала над признаками был произведен Z-scoring и данные были визуализированы.

В качестве группы объектов были выбраны те объекты, у которых Stress_Level больше среднего значения (они отмечены оранжевыми точками):



Как видно из графика, группа объектов находится в нижней части полуплоскости.

Далее над признаками была произведена стандартизация размахом, была выбрана группа объектов у которых значение признака Body_Temperature больше 0.2 раз своего максимума, и все было визуализировано:



Как видно из графика, группа объектов расположилась в верхней части координатной системы

Было использовано сингулярное разложение для Z-scored данных, в результате которого получилось два новых признака со следующими коэффициентами:

pca1 = 0.53* Body_Temperature + 0.46 * Stress_Level - 0.71 * Heart_Rate_Variability
pca2 = 0.66* Body_Temperature - 0.75 * Stress_Level + 0.004 * Heart_Rate_Variability

Также были посчитаны вклады соответственно каждой из них: 0.35, 0.33, 0.31.

Про PCA1: На этот компонент в первую очередь влияют температура тела (коэффициент: 0,53) и уровень стресса (коэффициент: 0,46).
Отрицательный вклад из Heart_Rate_Variability (-0,71) указывает на то, что по мере увеличения PCA1 вариабельность Heart_Rate_Variability снижается.
PCA1 может представлять собой общее физиологическое состояние, при котором

преобладают изменения температуры тела и уровня стресса, обратно связанные с вариабельностью сердечного ритма.
Про PCA2: На этот компонент сильно влияют температура тела (0,66) и уровень стресса (-0,75), оказывая противоположное влияние.
Очень низкий коэффициент вариабельности Heart_Rate_Variability (0,004) свидетельствует о том, что он оказывает минимальное влияние на PCA2.
PCA2 может отражать баланс или компромиссное решение между температурой тела и уровнем стресса, потенциально представляя собой взаимодействие стресса и терморегуляции.

Вывод: Первые два основных компонента (PCA1 и PCA2) в совокупности отражают большую часть различий в наборе данных, при этом температура тела и уровень стресса играют ведущую роль.
Вариабельность сердечного ритма в целом менее значима, но остается важной, особенно для PCA1.

Метод К-средних

Нами были выбраны следующие количественные признаки:

Sleep_Duration_Hours: Общее количество часов сна

Caffeine_Intake_mg: Количество потребления кофеина в миллиграммах

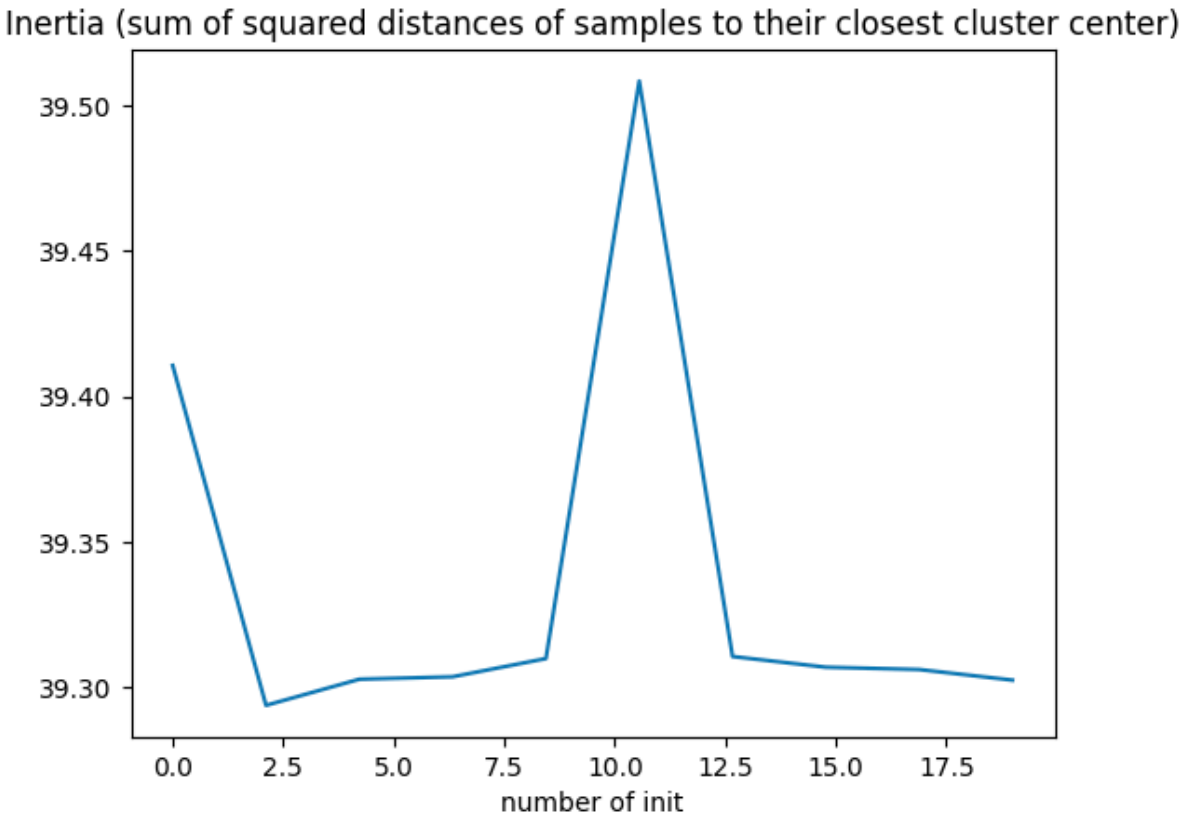
Heart_Rate_Variability: Изменчивость временных интервалов между ударами сердца

Выбор был обусловлен интересом анализа взаимосвязи данных признаков, которые оказывает влияние на качество сна.

Было рассмотрено разбиение на K=5 и K=9 признаков

K=5:

Вначале было сделано 10 случайных инициализаций, и посчитано среднее суммы квадратов дистанций до ближайшего кластерного центра:



После выбора удовлетворяющей минимуму критерия инициализации была построена таблица средних значений в кластерах:

Means	Sleep_Duration_Hours	Caffeine_Intake_mg	Heart_Rate_Variability
Cluster0	7.83	285.86	70.07
Cluster1	7.26	30.35	72.36
Cluster2	5.82	162.71	74.15
Cluster3	7.98	138.59	48.78
Cluster4	9.01	139.36	85.38
Grand mean	7.47	148.26	70.39

По приведенной таблице уже видно сильное различие кластеров. Например, Cluster4 имеет самая высокая Sleep_Duration_Hours и Heart_Rate_Variability, указывает на хорошо восстановившуюся и, возможно, заботящуюся о своем здоровье группу. У Cluster3 низкая Heart_Rate_Variability, несмотря на довольно длинный сон(Sleep_Duration_Hours). Из этого можно сделать вывод что метод точно работает.

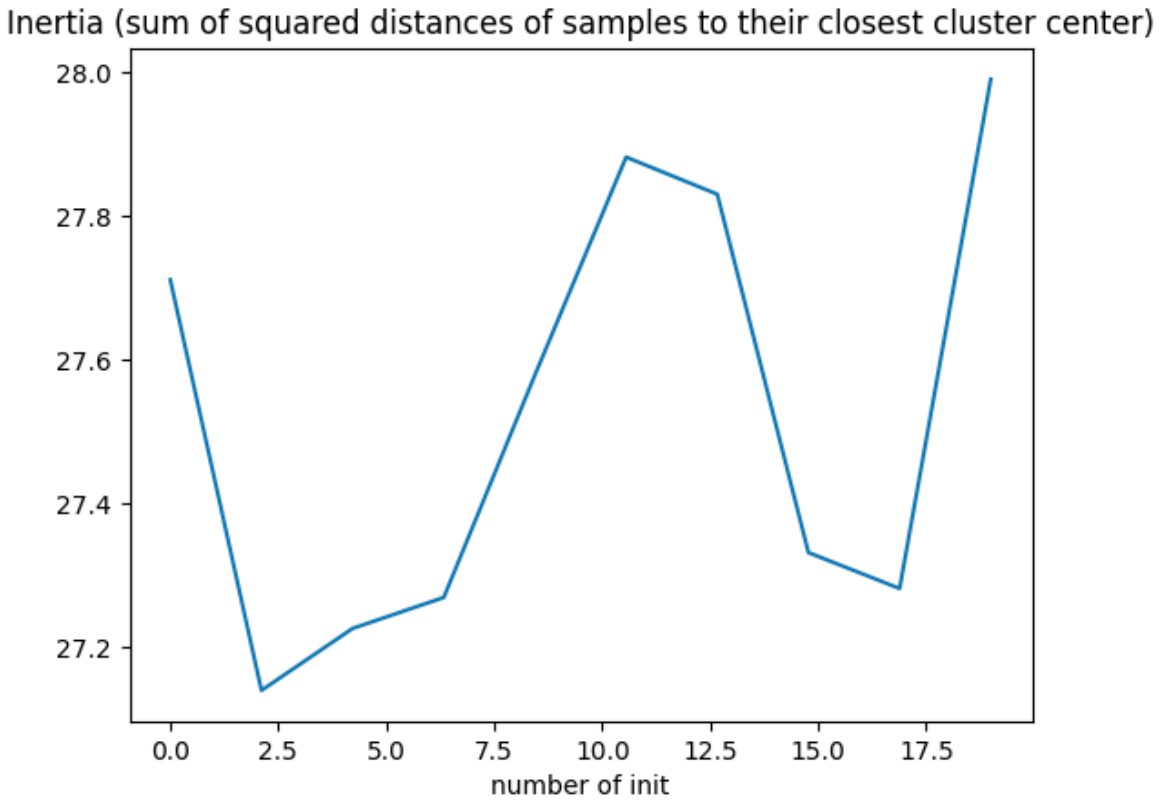
Далее были рассчитаны отклонения внутрикластерных средних от общих средних:

Rel.dif %	Sleep_Duration_Hours	Caffeine_Intake_mg	Heart_Rate_Variability	Number of objects
Cluster0	4.83	92.81	-0.46	195
Cluster1	-2.83	-79.53	2.80	228
Cluster2	-22.16	9.75	5.35	228
Cluster3	6.85	-6.52	-30.70	177
Cluster4	20.60	-6.00	21.31	172

Кластеризация выявила различия в образе жизни, а также заметные различия в том, как кофеин, сон и частота сердечных сокращений взаимодействуют между группами. Кластер 4 (длительный сон, высокая частота сердечных сокращений) и кластер 2 (короткий сон, хорошая частота сердечных сокращений) представляют особенно интересную физиологическую динамику.

K=9:

Вначале было сделано 10 случайных инициализаций, и посчитано среднее суммы квадратов дистанций до ближайшего кластерного центра:



После выбора удачной инициализации, была построена таблица средних значений по всем кластерам:

Means	Sleep_Duration_Hours	Caffeine_Intake_mg	Heart_Rate_Variability
Cluster0	5.44	216.31	68.27
Cluster1	8.05	168.48	52.08
Cluster2	9.62	235.37	73.75
Cluster3	9.26	87.49	84.92
Cluster4	7.54	316.36	68.77
Cluster5	6.46	29.63	85.98
Cluster6	5.87	107.74	62.32
Cluster7	7.19	173.51	90.32
Cluster8	8.15	39.10	54.19
Grand mean	7.47	148.26	70.39

По таблице видно явное различие между некоторыми кластерами. Кластеры с низким потреблением кофеина (например, кластер 3 и 5), как правило, имеют более высокую

частоту сердечных сокращений. Кластер 3 - оптимальная группа: Длительный сон, низкое содержание кофеина и высокий уровень HRV делают эту группу наиболее здоровой и сбалансированной. Для удобства дальнейшего анализа построим таблицу относительных отклонений внутрикластерных средних от общих средних:

Rel.dif %	Sleep_Duration_Hours	Caffeine_Intake_mg	Heart_Rate_Variability	Number of objects
Cluster0	-27.20	45.90	-3.01	96
Cluster1	7.79	13.64	-26.01	141
Cluster2	28.74	58.75	4.78	82
Cluster3	23.88	-40.99	20.64	104
Cluster4	0.88	113.38	-2.29	98
Cluster5	-13.58	-80.02	22.15	97
Cluster6	-21.45	-27.33	-11.46	126
Cluster7	-3.84	17.03	28.32	133
Cluster8	9.07	-73.63	-23.01	123

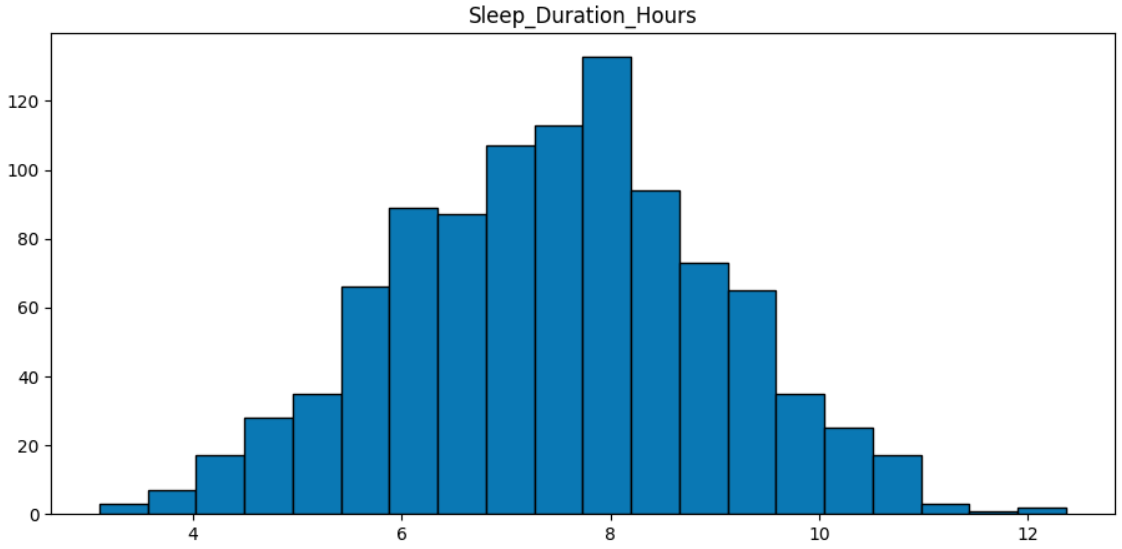
Кластеры 1 и 8 отличаются низким уровнем Heart_Rate_Variability, несмотря на сон выше среднего или низкое содержание кофеина. В отличие от этих кластеров кластер 3 с длительным сном, низким содержанием кофеина и высокая Heart_Rate_Variability делают группу 3 наиболее сбалансированной и здоровой.

Минимальное потребление кофеина (например, кластер 5) коррелирует с хорошим восстановлением, но высокое содержание кофеина (например, кластер 4) в некоторых случаях существенно не влияет на Heart_Rate_Variability, что свидетельствует об индивидуальной вариабельности. Кластер 4 и кластер 8 значительно отличаются по потреблению кофеина, но их продолжительность сна и Heart_Rate_Variability не сильно отличаются от среднего значения или других кластеров.

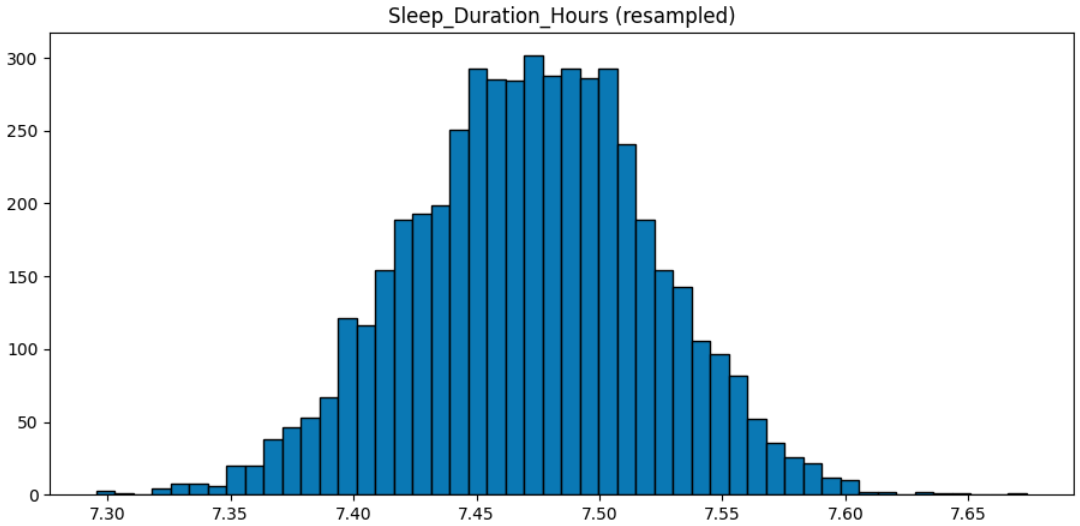
Вывод: k = 5 предлагает более простые и понятные кластеры, в то время как k = 9 может привести к чрезмерной сегментации данных, что приведет к избыточным или менее значимым группам

Бутстрэп

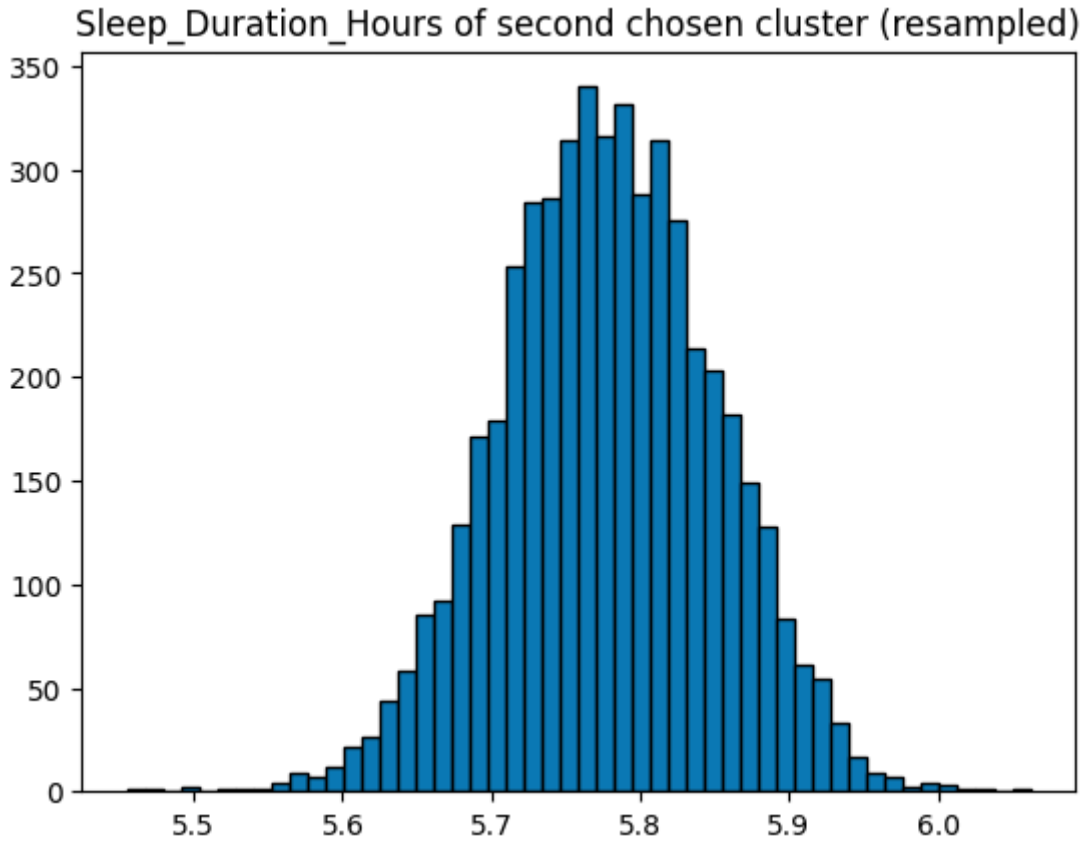
В этом задании авторами использовалось разбиение на 9 кластеров рассмотренное в предыдущем пункте отчета. Были построены 95% доверительные интервалы, проведено сравнение средних в двух кластерах и сравнение среднего на всем множестве и внутри кластера. Использовался метод бутстреп с опорой и без, в каждом применении бутстрепа было 5000 итераций. В качестве рассматриваемого признака был выбран Sleep_Duration_Hours со следующим распределением:



Заметим что у Sleep_Duration_Hours относительно близкое к нормальному распределение. Далее было построено распределение среднего значения величины Sleep_Duration_Hours на всей выборке:



И так же конкретном кластере **Cluster1**:



Заметим что видно смещение распределения. Что коррелирует с данными полученными в предыдущей главе работы.

Далее 95% доверительные интервалы для данных полученных бутстрепом на всем множестве объектов с использованием опоры и без:

Feature mean: 7.47

Confidence interval (pivotal): 7.38, 7.57

Confidence interval (non-pivotal): 7.37, 7.57

Разброс данных невелик, а доверительные границы практически совпадают. Это свидетельствует о высокой стабильности оценок, независимо от используемого метода бутстрепинга.

Рассмотрим 95% доверительный интервал для разности средних значений признака Sleep_Duration_Hours между кластерами Cluster0 и Cluster1 полученными в предыдущей главе работы:

Confidence interval (pivotal): 2.02, 2.53

Confidence interval (non-pivotal): 2.01, 2.54

Также рассмотрим 95% доверительный интервал для разности средних значений между Cluster1 и всеми объектами:

Confidence interval (pivotal): 0.37, 0.79

Confidence interval (non-pivotal): 0.37, 0.80

Выводы:

Значения полученные бутстрепом с и использованием опоры и без почти что совпадают. Данные результаты подтверждают, что метод кластеризации выделяет группы с реальными, статистически значимыми различиями, что является важным для интерпретации кластеров и дальнейшего использования результатов анализа.

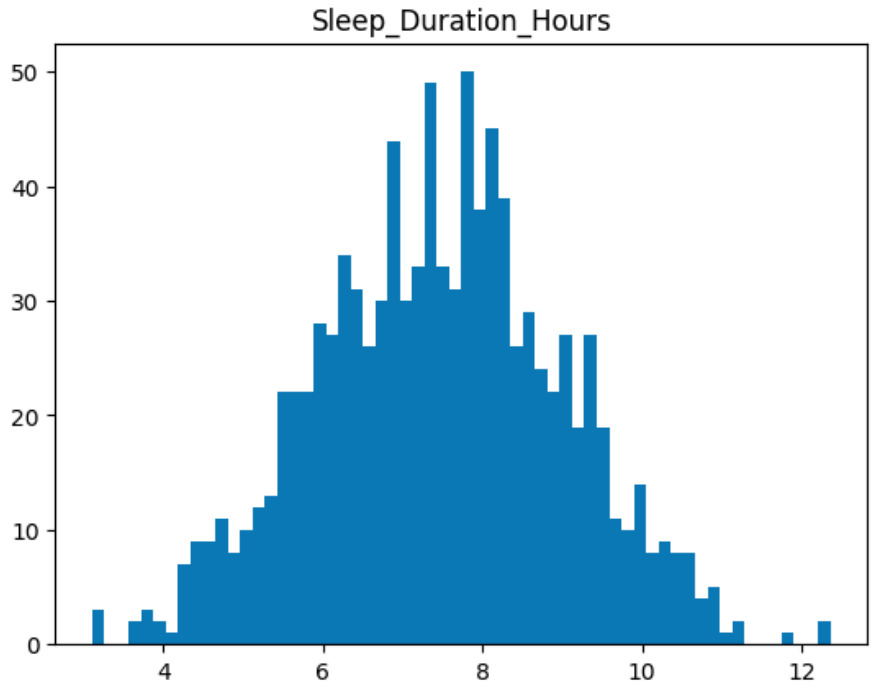
Отличие средних значений двух кластеров между собой и со всей выборкой подтверждают данные полученные в предыдущей главе работы.

Таблица сопряженности

Мы использовали следующие номинальные признаки: Sleep_Duration_Hours, Caffeine_Intake_mg, Stress_Level.

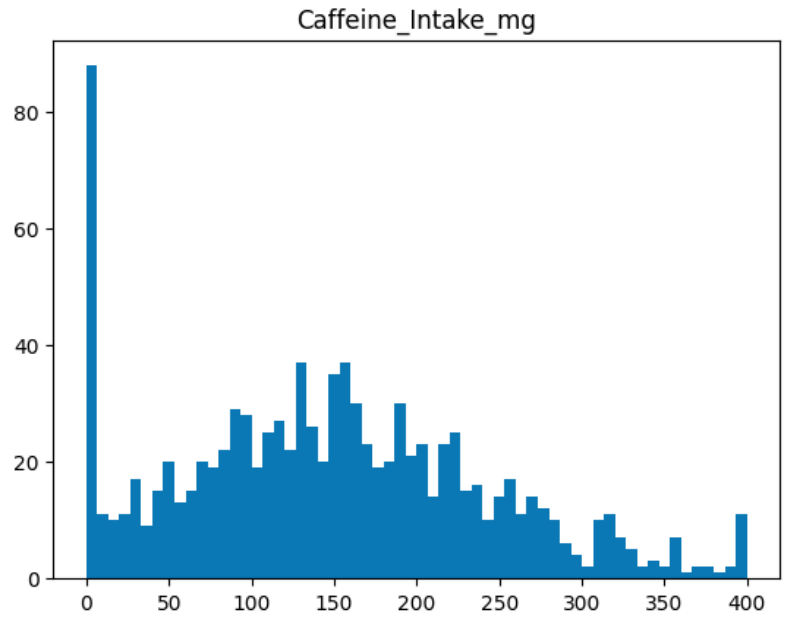
Сначала были построены гистограммы распределения трех базовых признаков:

Sleep_Duration_Hours



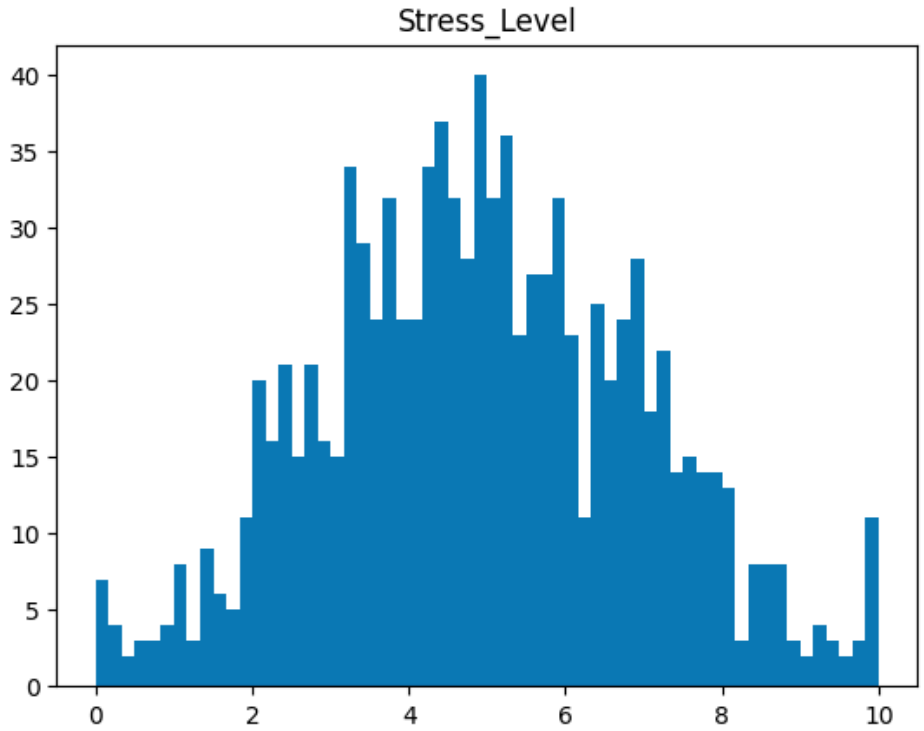
Видно, что здесь логично разбить базового признака на интервалы значений 1-6(low), 6-9 (medium), 9-13 (high)

Caffeine_Intake_mg



Будет логично, если выделим на интервалы 0-148(low), 148-300(medium), 300-400(high)

Stress_Level



Выберем 3 интервала: 0-3 (low), 3-6 (medium), 6-10(high)

Далее анализировалась взаимосвязь между Sleep_Duration_Hours и Caffeine_Intake_mg, а также между Sleep_Duration_Hours и Stress_Level. Были построены две таблицы сопряженности:

Sleep_Duration_Hours & Caffeine_Intake_mg:

Caffeine_Intake_mg	low	medium	high	All
Sleep_Duration_Hours				
low	92	76	6	174
medium	333	275	45	653
high	85	71	17	173
All	510	422	68	1000

Нормализованная Sleep_Duration_Hours & Caffeine_Intake_mg:

Caffeine_Intake_mg	low	medium	high	All
Sleep_Duration_Hours				
low	0.092	0.076	0.006	0.174
medium	0.333	0.275	0.045	0.653
high	0.085	0.071	0.017	0.173
All	0.510	0.422	0.068	1.000

Как видно из таблицы, средний сон и низкий кофеин - наиболее распространённое сочетание, вероятно, характеризующее основной профиль данных.
Люди с высокими уровнями потребления кофеина (особенно при низком и высоком уровне сна) встречаются редко, всего в 6.8%. Что согласуется со здравым смыслом.

Sleep_Duration_Hours & Stress_Level:

Stress_Level	low	medium	high	All
Sleep_Duration_Hours				
low	28	92	54	174
medium	118	338	197	653
high	28	100	45	173
All	174	530	296	1000

Нормализованная Sleep_Duration_Hours & Stress_Level:

Stress_Level	low	medium	high	All
Sleep_Duration_Hours				
low	0.028	0.092	0.054	0.174
medium	0.118	0.338	0.197	0.653
high	0.028	0.100	0.045	0.173
All	0.174	0.530	0.296	1.000

Как видно из таблицы, больше людей имеют высокий стресс, чем у тех есть низкий. Более того, большая часть людей имеет сон средней длительность под стрессом

среднего или высокого(**33.8%** и **19.7%**). Не мало людей со средним уровнем (**9.2%**) мало спит тоже.

Далее были построены матрицы условных вероятностей:

Sleep_Duration_Hours & Caffeine_Intake_mg:

Caffeine_Intake_mg	low	medium	high	All
Sleep_Duration_Hours				
low	0.1804	0.1801	0.0882	0.174
medium	0.6529	0.6517	0.6618	0.653
high	0.1667	0.1682	0.2500	0.173
All	1.0000	1.0000	1.0000	1.000

По таблице можно сделать вывод, что весьма высока вероятность(**~65%**) того что люди, которые употребляют кофеин будут иметь сон средней длительность. Более того, есть вероятность **25%** что люди, которые много употребляют кофеин долго спат.

Sleep_Duration_Hours & Stress_Level:

Stress_Level	low	medium	high	All
Sleep_Duration_Hours				
low	0.1609	0.1736	0.1824	0.174
medium	0.6782	0.6377	0.6655	0.653
high	0.1609	0.1887	0.1520	0.173
All	1.0000	1.0000	1.0000	1.000

Из матрицы условных вероятностей можно сделать вывод, что при любом уровне стресса вероятность средней длительности сна значительно выше, чем вероятности низкого или высокого сна. Помимо этого, высокий стресс чаще ассоциируется с низкой продолжительностью сна

Были построены **матрицы индексов Кетле (Relative Quetelet Indexes):**

Sleep_Duration_Hours & Caffeine_Intake_mg:

Caffeine_Intake_mg	low	medium	high
Sleep_Duration_Hours			
low	0.036737	0.035028	-0.492901
medium	-0.000090	-0.002054	0.013422
high	-0.036609	-0.027477	0.445087

Sleep_Duration_Hours & Stress_Level:

Stress_Level	low	medium	high
Sleep_Duration_Hours			
low	-0.075175	-0.002386	0.048462
medium	0.038531	-0.023375	0.019205
high	-0.069829	0.090631	-0.121231

Заметим, что:

Средний уровень стресса способствует большей продолжительности сна, тогда как высокий стресс связан с более коротким сном.

Участники с высоким уровнем потребления кофеина демонстрируют две крайности: высокая продолжительность сна (положительный индекс) или её отсутствие (отрицательный индекс для низкого сна).

При расчете среднего индекса Кетле:

Sleep_Duration_Hours & Caffeine_Intake_mg: 0.0056

Sleep_Duration_Hours & Stress_Level: 0.0024

Видим большую связь между длительностью сна и потреблением кофеина, чем между уровнем стресса и длительностью сна.

Объяснение:

Таблицы с кофеином показывают более значительные отклонения частот (как положительные, так и отрицательные) в индексе Кетле. Например, высокий кофеин сильно связан с высокой продолжительностью сна (+0.445) и с низкой продолжительностью сна (-0.493).

В таблице с уровнем стресса индексы Кетле менее выражены, что говорит о более

слабом влиянии уровня стресса на распределение продолжительности сна.

При расчете коэффициент хи-квадрат:

Sleep_Duration_Hours & Caffeine_Intake_mg: 0.0056

Sleep_Duration_Hours & Stress_Level: 0.0024

Оказалось, что значения коэффициента хи-квадрат для обеих таблиц равны среднему индексу Кетле. Они равны, так как в основе их расчета лежит одна и та же концепция измерения отклонений наблюдаемых частот от ожидаемых в таблице сопряженности. Доказательство было найдено в “Eleven Ways to Look at the Chi-Squared Coefficient for Contingency Tables, Boris Mirkin”, п. 3.6 “Chi-Squared as the Average Relative Quetelet Index”.

Приложение 1 (Линейная регрессия и коэффициент корреляции)

1

```
df1 = df.filter(['Sleep_Quality_Score', 'Caffeine_Intake_mg','Light_Exposure_hours'])
pd.plotting.scatter_matrix(df1, alpha=0.5, diagonal='kde', figsize=(8,8))
```

2)

```
plt.figure(figsize=(10, 6))
plt.scatter(df['Caffeine_Intake_mg'], df['Sleep_Quality_Score'])
plt.ylabel('Sleep_Quality_Score')
plt.xlabel('Caffeine_Intake_mg')
plt.grid()
```

3)

```
a = np.sum((y - y.mean()) * X) / np.sum((X - X.mean()) * X)
b = y.mean() - a * X.mean()
print('Regression coefficient: %.5f' % a)
print('Intercept: %.5f' % b)
```

4)

```
from sklearn.metrics import r2_score

corr_matrix = np.corrcoef(X, y)
r0 = corr_matrix[0][1]
print('Correlation coefficient: %.3f' % r0)
X_reshaped = X.values.reshape(-1, 1)

y_pred = model.predict(X_reshaped)
r2 = r2_score(y, y_pred)
print('Determination coefficient: %.3f' % r2)
```

5)

```
X_cut_reshaped = X_cut.values.reshape(-1, 1)

y_pred_1 = model_1.predict(X_cut_reshaped)
r2_1 = r2_score(y_cut, y_pred_1)
print('Determination coefficient: %.3f' % r2_1)
```

6)

```
N = 4

ix = np.random.choice(np.arange(N), 4, replace=False)
x_ix = X_reshaped[ix]
y_true_ix = y[ix]
y_pred_ix = y_pred[ix]

df_rand_obj = pd.DataFrame({
    'x_ix': x_ix.ravel(),
    'y_true_ix': y_true_ix.ravel() ,
    'y_pred_ix': y_pred_ix.ravel(),
})
df_rand_obj
```

7)

```
d = abs(y - y_pred)
dda = d/abs(y)
dml = d/abs(y_pred)
eda = np.mean(dda)
eml = np.mean(dml)


eda, eml
```

Приложение 2 (Метод главных компонент)

1)

```
Xz = (df_dz2 - df_dz2.mean()) / df_dz2.std()
uz, sz, vhz = np.linalg.svd(Xz)

uz0 = -uz[:, 0]
vhz0 = -vhz[0]
uz1 = uz[:, 1]

plt.scatter(uz0, uz1)
plt.scatter(uz0[Xz.iloc[:, 1].values > Xz.iloc[:, 1].mean()], uz1[Xz.iloc[:, 1].values > Xz.iloc[:, 1].mean()])
plt.title('PCA of Z-score standardized data')
plt.show()
```

2)

```
Xr = (df_dz2 - df_dz2.mean()) / (df_dz2.max() - df_dz2.min())
ur, sr, vhr = np.linalg.svd(Xr)

ur0 = -uz[:, 0]
vhr0 = -vhz[0]
ur1 = uz[:, 1]

plt.scatter(ur0, ur1)
plt.scatter(ur0[Xr.iloc[:, 0].values > 0.2 * Xr.iloc[:, 0].max()], ur1[Xr.iloc[:, 0].values > 0.2 * Xr.iloc[:, 0].max()])
plt.title('PCA of range standardized data')
plt.show()
```

3)

```
pca1 = vhz[0]
pca2 = vhz[1]
```

4)

```
data_scatter = np.sum((Xz**2).values)
contributions = sz**2 /data_scatter
```

Приложение 3 (Метод К-средних)

1)

```
from sklearn.cluster import KMeans


def make_clusters(data, n_clusters):
 inertias = []
 labels = []

 for i in range(10):
     kmeans = KMeans(n_clusters=n_clusters, init='random', n_init=1)
     kmeans.fit(data)
     inertias.append(kmeans.inertia_)
     labels.append(kmeans.labels_)

 x = np.linspace(0, 19, 10)
 plt.plot(x, inertias)
 plt.xlabel('number of init')
 plt.title('Inertia (sum of squared distances of samples to their closest cluster center)')
 plt.show()

 return inertias, labels
```

```
df_stand = df[['Sleep_Duration_Hours', 'Caffeine_Intake_mg', 'Heart_Rate_Variability']]
df_stand = (df_stand - df_stand.mean()) / (df_stand.max() - df_stand.min())
inertias_5, labels_5 = make_clusters(df_stand, 5)
```

2)

```
def calculate_means(inertias, labels, n_clusters, data_stand,
data_full):
    ind_min = inertias.index(min(inertias))
    label = labels[ind_min]

    clusters = {}
    cluster_means = []

    for k in range(n_clusters):
        clusters['Cluster' + str(k)] = data_full.values[np.where(label== k)]

    for name_cluster in clusters:
        cluster_means.append(np.mean(clusters[name_cluster], axis=0))

    grand_mean = np.mean(data_full, axis=0).values

    means = pd.DataFrame(
    (cluster_means + [grand_mean]),
    (list(clusters.keys()) + ['Grand mean']),
    data_stand.columns
    )
    means.index.name = 'Means'

    num_objects = [len(v) for k, v in clusters.items()]

    relative_differences = 100 * np.divide(np.subtract(cluster_means, grand_mean), grand_mean)

    rel_dif = pd.DataFrame(
    relative_differences,
    clusters.keys(),
    data_stand.columns
    )
    rel_dif['Number of objects'] = num_objects
    rel_dif.index.name = 'Rel.dif %'

    return means.round(2), rel_dif.round(2), label

m5, rdiff5, label = calculate_means(inertias_5, labels_5, 5, df_stand,
df_full)
m5
```

Приложение 4 (Бутстрэп)

1)

```
feature = df.iloc[:, 3]
# визуализация
plt.figure(figsize=(11,5))
plt.hist(feature, bins=20, edgecolor='black')
plt.title('Sleep_Duration_Hours')
plt.show()
```

2)

```
n_bootstrap = 5000
# бутстреп
r = np.random.choice(feature, size=(len(feature), n_bootstrap))
rm = np.mean(r, axis=0)
# визуализация
plt.figure(figsize=(11,5))
plt.hist(rm, bins=50, edgecolor='black')
plt.title('Sleep_Duration_Hours (resampled)')
plt.show()
```

3)

```
fid1, fid2 = 0, 1
feature_1 = feature.values[np.where(label == fid1)]
# средние
r1 = r * np.isin(r, feature_1)
rm1 = np.sum(r1, axis=0) / np.count_nonzero(r1, axis=0)
rm1 = np.nan_to_num(rm1, nan=np.mean(rm1[~np.isnan(rm1)]))

feature_2 = feature.values[np.where(label == fid2)]
r2 = r * np.isin(r, feature_2)
rm2 = np.sum(r2, axis=0) / np.count_nonzero(r2, axis=0)
rm2 = np.nan_to_num(rm2, nan=np.mean(rm2[~np.isnan(rm2)]))

plt.hist(rm2, bins=50, edgecolor='black')
plt.title('Sleep_Duration_Hours of second chosen cluster (resampled)')
plt.show()
```

4)

```
def bootstrap(x, n_bootstrap=n_bootstrap):
 # pivotal
 mean_p = np.mean(x)
 std_p = np.std(x)
 lbord_p = mean_p - 1.96 * std_p
 rbord_p = mean_p + 1.96 * std_p
 confint_p = np.array([lbord_p, rbord_p])

 # non-pivotal
 x_sorted = np.sort(x)
 lbord_np = x_sorted[int(n_bootstrap * 0.025) + 1]
 rbord_np = x_sorted[int(n_bootstrap * 0.975)]
 confint_np = np.array([lbord_np, rbord_np])

 return confint_p, confint_np
# вывод
rm_confint_p, rm_confint_np = bootstrap(rm)
print('Feature mean: %.2f' % np.mean(feature))
print('Confidence interval (pivotal): %.2f, %.2f' % (rm_confint_p[0], rm_confint_p[1]))
print('Confidence interval (non-pivotal): %.2f, %.2f' % (rm_confint_np[0], rm_confint_np[1]))
```

5)

```
rm1_rm2_confint_p, rm1_rm2_confint_np = bootstrap(rm1 - rm2)
print('Confidence interval (pivotal): %.2f, %.2f' %
(rm1_rm2_confint_p[0], rm1_rm2_confint_p[1]))
print('Confidence interval (non-pivotal): %.2f, %.2f' %
(rm1_rm2_confint_np[0], rm1_rm2_confint_np[1]))
```

6)

```
rm1_rm_confint_p, rm1_rm_confint_np = bootstrap(rm1 - rm)
print('Confidence interval (pivotal): %.2f, %.2f' %
(rm1_rm_confint_p[0], rm1_rm_confint_p[1]))
print('Confidence interval (nonpivotal): %.2f, %.2f' %
(rm1_rm_confint_np[0], rm1_rm_confint_np[1]))
```

Приложение 5 (Таблица сопряженности)

1)

```
crosstab_1 = pd.crosstab(df_new['Sleep_Duration_Hours'], df_new['Caffeine_Intake_mg'], margins=True)
crosstab_1
```

2)

```
relfreq_1 = pd.crosstab(df_new['Sleep_Duration_Hours'], df_new['Caffeine_Intake_mg'], margins=True, normalize=True)
relfreq_1.round(4)
```

3)

```
crosstab_2 = pd.crosstab(df_new['Sleep_Duration_Hours'], df_new['Stress_Level'], margins=True)
crosstab_2
```

4)

```
relfreq_2 = pd.crosstab(df_new['Sleep_Duration_Hours'], df_new['Stress_Level'], margins=True, normalize=True)
relfreq_2.round(4)
```

5)

```
condfreq_1 = np.divide(crosstab_1, crosstab_1[-1:])
condfreq_1.round(4)
```

6)

```
condfreq_2 = np.divide(crosstab_2, crosstab_2[-1:])
condfreq_2.round(4)
```

7)

```
relfreq_indep_1 = np.dot(relfreq_1.iloc[:,-1].values[:,None], relfreq_1.iloc[-1,:].values[None,:])

quetelet_1 = relfreq_1 / relfreq_indep_1 - 1 # матрица Кетле
quetelet_1 = quetelet_1.iloc[:-1, :-1]
quetelet_1.round(4)
m_1 = relfreq_1.iloc[:-1, :-1] * quetelet_1
```

8)

```
relfreq_indep_2 = np.dot(relfreq_2.iloc[:,-1].values[:,None], relfreq_2.iloc[-1,:].values[None,:])

quetelet_2 = relfreq_2 / relfreq_indep_2 - 1 # матрица Кетле
quetelet_2 = quetelet_2.iloc[:-1, :-1]
quetelet_2.round(4)
m_2 = relfreq_2.iloc[:-1, :-1] * quetelet_2
np.sum(m_2.values)
```

9)

```
h_1 = (relfreq_1 - relfreq_indep_1)**2 / relfreq_indep_1
h_1 = h_1.iloc[:-1, :-1]
h_1.round(4)
np.sum(h_1.values)

h_2 = (relfreq_2 - relfreq_indep_2)**2 / relfreq_indep_2
h_2 = h_2.iloc[:-1, :-1]
h_2.round(4)
np.sum(h_2.values)
```